NP-Vollständigkeit des Erfüllbarkeitsproblems

Prof. Dr. Berthold Vöcking
Lehrstuhl Informatik 1
Algorithmen und Komplexität
RWTH Aachen

Dezember 2011

Def: NP-Härte

Definition (NP-Härte)

Ein Problem L heißt NP*-hart, wenn* $\forall L' \in \mathrm{NP} : L' \leq_p L$.

Satz

L NP*-hart*, $L \in \mathrm{P} \Rightarrow \mathrm{P} = \mathrm{NP}$

Beweis: Polyzeitalgo für L liefert Polyzeitalgo für alle $L' \in \mathrm{NP}$. □

Fazit: NP-harte Probleme haben keine Polyzeitalgo, es sei denn P= NP.

Def: NP-Vollständigkeit

Definition (NP-Vollständigkeit)

Ein Problem L heißt NP-vollständig, *falls gilt*

1. $L \in$ NP, *und*
2. L *ist* NP*-hart.*

Die Klasse der NP*-vollständigen Probleme wird mit* NPC *bezeichnet.*

Wir werden zeigen, dass SAT, CLIQUE, KP-E, BPP-E, TSP-E und viele weitere Probleme NP-vollständig sind.

Keines dieser Probleme hat somit einen Polynomialzeitalgorithmus; es sei denn P$=$ NP.

NP-Vollständigkeit des Erfüllbarkeitsproblems

Der Ausgangspunkt für unsere NP-Vollständigkeitsbeweise ist das Erfüllbarkeitsproblem.

Satz (Cook und Levin)

SAT ist NP*-vollständig.*

SAT hat somit keinen Polynomialzeitalgorithmus; es sei denn $P = NP$.

Beweis des Satzes von Cook und Levin

Offensichtlich gilt SAT $\in$ NP, denn die erfüllende Belegung kann als Zertifikat verwendet werden. Wir müssen also „nur“ noch zeigen, dass SAT NP-hart ist.

Sei $L \subseteq \Sigma^*$ ein Problem aus NP. Wir müssen zeigen $L \leq_p$ SAT.

Dazu konstruieren wir eine polynomiell berechenbare Funktion f, die jedes $x \in \Sigma^*$ auf eine Formel ϕ abbildet, so dass gilt

$$x \in L \;\Leftrightarrow\; \phi \in SAT \;\; .$$

Beweis des Satzes von Cook und Levin

M sei eine NTM, die L in polynomieller Zeit erkennt. Wir zeigen

$$M \text{ akzeptiert } x \Leftrightarrow \phi \in SAT \ .$$

Eigenschaften von M

- O.B.d.A. besuche M keine Bandpositionen links von der Startposition.
- Eine akzeptierende Rechnung von M gehe in den Zustand q_{accept} über und bleibt dort in einer Endlosschleife.
- Sei $p(\cdot)$ ein Polynom, so dass M eine Eingabe x genau dann akzeptiert, wenn es einen Rechenweg gibt, der nach $p(n)$ Schritten im Zustand q_{accept} ist, wobei n die Länge von x bezeichne.

Beweis des Satzes von Cook und Levin

Beobachtung:

Sei $K_0 = q_0 x$ die Startkonfiguration von M. M akzeptiert genau dann, wenn es einen Rechenweg, d.h. eine mögliche Konfigurationsfolge

$$K_0 \vdash K_1 \vdash \cdots \vdash K_{p(n)}$$

gibt, bei der $K_{p(n)}$ im Zustand q_{accept} ist.

Weiteres Vorgehen:

Wir konstruieren die Formel ϕ derart, dass ϕ genau dann erfüllbar ist, wenn es eine solche akzeptierende Konfigurationsfolge gibt.

Beweis des Satzes von Cook und Levin

Variablen in ϕ

- $Q(t,k)$ für $t \in \{0,\ldots,p(n)\}$ und $k \in Q$
- $H(t,j)$ für $t,j \in \{0,\ldots,p(n)\}$
- $S(t,j,a)$ für $t,j \in \{0,\ldots,p(n)\}$ und $a \in \Gamma$

Interpretation der Variablen:

- Die Belegung $Q(t,k) = 1$ soll besagen, dass sich die Rechnung zum Zeitpunkt t im Zustand k befindet.
- Die Belegung $H(t,j) = 1$ steht dafür, dass sich der Kopf zum Zeitpunkt t an Bandposition j befindet.
- die Belegung $S(t,j,a) = 1$ bedeutet, dass zum Zeitpunkt t an Bandposition j das Zeichen a geschrieben steht.

Beweis des Satzes von Cook und Levin

Kodierung einzelner Konfigurationen in der Teilformel ϕ_t:

Für jedes $t \in \{0, \ldots, p(n)\}$, benötigen wir eine Formel ϕ_t, die nur dann erfüllt ist, wenn es

1. genau einen Zustand $k \in Q$ mit $Q(t,k) = 1$ gibt,
2. genau eine Bandposition $j \in \{0, \ldots, p(n)\}$ mit $H(t,j) = 1$ gibt, und
3. für jedes $j \in \{0, \ldots, p(n)\}$ jeweils genau ein Zeichen $a \in \Gamma$ mit $S(t,j,a) = 1$ gibt.

Beweis des Satzes von Cook und Levin

Erläuterung zur Formel ϕ_t:

- Für eine beliebige Variablenmenge $\{y_1, \ldots, y_m\}$ besagt das folgende Prädikat in KNF, dass genau eine der Variablen y_i den Wert 1 annimmt:

$$(y_1 \vee \ldots \vee y_m) \wedge \bigwedge_{i \neq j} (\bar{y}_i \vee \bar{y}_j)$$

- Die Anzahl der Literale in dieser Formel ist quadratisch in der Anzahl der Variablen.
- Die drei Anforderungen können also jeweils durch eine Formel in polynomiell beschränkter Länge kodiert werden.

Wir betrachten nun nur noch Belegungen, die die Teilformeln $\phi_0, \ldots, \phi_{p(n)}$ erfüllen und somit Konfigurationen $K_0, \ldots, K_{p(n)}$ beschreiben.

Beweis des Satzes von Cook und Levin

Als nächstes konstruieren wir eine Formel ϕ'_t für $1 \leq t \leq p(n)$, die nur für solche Belegungen erfüllt ist, bei denen K_t eine direkte Nachfolgekonfiguration von K_{t-1} ist.

Die Formel ϕ'_t kodiert zwei Eigenschaften:

1. Die Bandinschrift von K_t stimmt an allen Positionen außer der Kopfposition (zum Zeitpunkt $t-1$) mit der Inschrift von K_{t-1} überein.
2. Zustand, Kopfposition und Bandinschrift an der Kopfposition verändern sich gemäß der Übergangsrelation δ.

Beweis des Satzes von Cook und Levin

Die Eigenschaft, dass die Bandinschrift von K_t an allen Positionen außer der Kopfposition (zum Zeitpunkt $t-1$) mit der Inschrift von K_{t-1} übereinstimmt, kann wie folgt kodiert werden:

$$\bigwedge_{i=0}^{p(n)} \bigwedge_{z\in\Gamma} ((S(t-1,i,z) \wedge \neg H(t-1,i)) \Rightarrow S(t,i,z))$$

Dabei steht $A \Rightarrow B$ für $\neg A \vee B$. D.h. die Formel lautet eigentlich

$$\bigwedge_{i=0}^{p(n)} \bigwedge_{z\in\Gamma} (\neg(S(t-1,i,z) \wedge \neg H(t-1,i)) \vee S(t,i,z))$$

Das *De Morgansche Gesetz* besagt, dass $\neg(A \wedge B)$ äquivalent ist zu $\neg A \vee \neg B$. Dadurch ergibt sich folgende Teilformel in KNF

$$\bigwedge_{i=0}^{p(n)} \bigwedge_{z\in\Gamma} (\neg S(t-1,i,z) \vee H(t-1,i) \vee S(t,i,z))$$

Beweis des Satzes von Cook und Levin

Für die Eigenschaft, dass Zustand, Kopfposition und Bandinschrift an der Kopfposition sich gemäß der Ubergangsrelation δ verändern, ergänzen wir für alle $k \in Q$, $j \in \{0, \ldots, p(n)-1\}$ und $a \in \Gamma$ die folgende Teilformel

$$(Q(t-1,k) \wedge H(t-1,j) \wedge S(t-1,j,a)) \quad \Rightarrow \bigvee_{(k,a,k',a',\kappa)\in\delta} (Q(t,k') \wedge H(t,j+\kappa) \wedge S(t,j,a')) \ ,$$

wobei κ die Werte $L=-1$, $N=0$ und $R=1$ annehmen kann.

Wie lässt sich diese Teilformel in die KNF transformieren?

Beweis des Satzes von Cook und Levin

Ersetzen von $A \Rightarrow B$ durch $\neg A \vee B$ und Anwenden des De Morganschen Gesetzes ergibt die Teilformel

$$\neg Q(t-1,k) \vee \neg H(t-1,j) \vee \neg S(t-1,j,a)) \vee \bigvee_{(k,a,k',a',\kappa)\in\delta} (Q(t,k') \wedge H(t,j+\kappa) \wedge S(t,j,a')) \ ,$$

wobei κ die Werte $L = -1$, $N = 0$ und $R = 1$ annehmen kann.

Jetzt müssen noch die inneren $\wedge$-Verknüpfungen „ausmultipliziert“ werden, d.h. wir ersetzten wiederholt eine Formel der Form $X \vee (A \wedge B \wedge C)$ durch eine äquivalente Formel $(X \vee A) \wedge (X \vee B) \wedge (X \vee C)$ bis wir eine Formel in KNF erhalten.

Damit ist die Beschreibung von ϕ'_t abgeschlossen.

Beweis des Satzes von Cook und Levin

Die Gesamtformel ϕ ergibt sich nun wie folgt:

$$Q(0,q_0) \wedge H(0,0) \wedge \bigwedge_{i=0}^{n} S(0,i,x_i) \wedge \bigwedge_{i=n+1}^{p(n)} S(0,i,B)$$
$$\wedge \bigwedge_{i=0}^{p(n)} \phi_i \wedge \bigwedge_{i=1}^{p(n)} \phi'_i \wedge Q(p(n), q_{\text{accept}})$$

Die Länge von ϕ ist polynomiell beschränkt in n, und ϕ ist effizient aus x berechenbar.

Gemäß unserer Konstruktion ist ϕ genau dann erfüllbar, wenn es eine akzeptierende Konfigurationsfolge für M auf x der Länge $p(n)$ gibt. □

Kochrezept für NP-Vollständigkeitsbeweise

- Um Nachzuweisen, dass SAT NP-hart ist, haben wir in einer „Master-Reduktion“ alle Probleme aus NP auf SAT reduziert.
- Die NP-Vollständigkeit von SAT können wir jetzt verwenden, um nachzuweisen, dass weitere Probleme NP-hart sind.

Lemma

L^* NP-*hart*, $L^* \leq_p L \Rightarrow L$ NP-*hart*.

Beweis: Gemäß Voraussetzung gilt $\forall L' \in \text{NP} : L' \leq_p L^*$ und $L^* \leq_p L$. Aufgrund der Transitivität der polynomiellen Reduktion folgt somit $\forall L' \in \text{NP} : L' \leq_p L$. □

NP-Vollständigkeit von 3SAT

Eine Formel in *k-KNF* besteht nur aus Klauseln mit jeweils k Literalen, sogenannten *k-Klauseln*.

Problem (3SAT)

Eingabe: Aussagenlogische Formel ϕ in 3-KNF
Frage: Gibt es eine erfüllende Belegung für ϕ?

- 3SAT ist ein Spezialfall von SAT und deshalb wie SAT in NP.
- Um zu zeigen, dass 3SAT ebenfalls NP-vollständig ist, müssen wir also nur noch die NP-Härte von 3SAT nachweisen.
- Dazu zeigen wir SAT $\leq_p$ 3SAT.

SAT $\leq_p$ 3SAT

Lemma

SAT $\leq_p$ 3SAT.

Beweis:

- Gegeben sei eine Formel ϕ in KNF.
- Wir transformieren ϕ in eine *erfüllbarkeitsäquivalente* Formel ϕ' in 3KNF, d.h.

$$\phi \text{ ist erfüllbar } \Leftrightarrow \phi' \text{ ist erfüllbar .}$$

- Aus einer 1- bzw 2-Klausel können wir leicht eine äquivalente 3-Klausel machen, indem wir ein Literal wiederholen.
- Was machen wir aber mit k-Klauseln für $k > 3$?

SAT $\leq_p$ 3SAT

- Sei $k \geq 4$ und C eine k-Klausel der Form

$$C = \ell_1 \vee \ell_2 \vee \ell_3 \cdots \vee \ell_k \ .$$

- In einer *Klauseltransformation* ersetzen wir C durch die Teilformel

$$C' = (\ell_1 \vee \cdots \vee \ell_{k-2} \vee h) \wedge (\bar{h} \vee \ell_{k-1} \vee \ell_k) \ ,$$

wobei h eine zusätzlich eingeführte Hilfsvariable bezeichnet.

Beispiel für die Klauseltransformation:
Aus der 5 Klausel

$$x_1 \vee \bar{x}_2 \vee x_3 \vee x_4 \vee \bar{x}_5$$

wird in einem ersten Transformationsschritt die Teilformel

$$(x_1 \vee \bar{x}_2 \vee x_3 \vee h_1) \ \wedge \ (\bar{h}_1 \vee x_4 \vee \bar{x}_5) \ ,$$

also eine 4- und eine 3-Klausel. Auf die 4-Klausel wird die Transformation erneut angewandt. Wir erhalten die Teilformel

$$(x_1 \vee \bar{x}_2 \vee h_2) \ \wedge \ (\bar{h}_2 \vee x_3 \vee h_1) \ \wedge \ (\bar{h}_1 \vee x_4 \vee \bar{x}_5) \ ,$$

die nur noch 3-Klauseln enthält.

Nachweis der Erfüllbarkeitsäquivalenz:

ϕ' sei aus ϕ entstanden durch Ersetzen von C durch C'.

zz: ϕ erfüllbar $\Rightarrow$ ϕ' erfüllbar

- Sei B eine erfüllende Belegung für ϕ.
- B weist mindestens einem Literal aus C hat den Wert 1 zu.
- Wir unterscheiden zwei Fälle:
  1) Falls $\ell_1 \vee \cdots \vee \ell_{k-2}$ erfüllt ist, so ist ϕ' erfüllt, wenn wir $h = 0$ setzen.
  2) Falls $\ell_{k-1} \vee \ell_k$ erfüllt ist, so ist ϕ' erfüllt, wenn wir $h = 1$ setzen.
- Also ist ϕ' in beiden Fällen erfüllbar.

SAT $\leq_p$ 3SAT

zz: ϕ' erfüllbar $\Rightarrow$ ϕ erfüllbar

- Sei B nun eine erfüllende Belegung für ϕ'.
- Wir unterscheiden wiederum zwei Fälle:
  - Falls B der Variable h den Wert 0 zuweist, so muss B einem der Literale $\ell_1, \ldots, \ell_{k-2}$ den Wert 1 zugewiesen haben.
  - Falls B der Variable h den Wert 0 zuweist, so muss B einem der beiden Literale ℓ_3 oder ℓ_4 den Wert 1 zugewiesen haben.
- In beiden Fällen erfüllt B somit auch ϕ.

SAT $\leq_p$ 3SAT

- Durch Anwendung der Klauseltransformation entstehen aus einer k-Klausel eine $(k-1)$-Klausel und eine 3-Klausel.
- Nach $k-3$ Iterationen sind aus einer k Klausel somit $k-2$ viele 3-Klauseln entstanden.
- Diese Transformation wird solange auf die eingegebene Formel ϕ angewandt, bis die Formel nur noch 3-Klauseln enthält.
- Wenn n die Anzahl der Literale in ϕ ist, so werden insgesamt höchstens $n-3$ Klauseltransformationen benötigt.
- Die Laufzeit ist somit polynomiell beschränkt.

□

Korollar

3SAT ist NP*-vollständig.*

Karps Liste mit 21 NP-vollständigen Problemen

- SAT
  - CLIQUE
  - SET PACKING
  - VERTEX COVER
    - SET COVERING
    - FEEDBACK ARC SET
    - FEEDBACK NODE SET
    - DIRECTED HAMILTONIAN CIRCUIT
      - UNDIRECTED HAMILTONIAN CIRCUIT

Die Schachtelungstiefe beschreibt den Weg der Reduktionen, wie Karp sie in seinem Artikel geführt hat.

Karps Liste mit 21 NP-vollständigen Problemen

- SAT
  - 0-1 INTEGER PROGRAMMING
  - 3SAT
    - CHROMATIC NUMBER (COLORING)
      - CLIQUE COVER
      - EXACT COVER
        - 3-dimensional MATCHING
        - STEINER TREE
        - HITTING SET
        - KNAPSACK
          - JOB SEQUENCING
          - PARTITION
            - MAX-CUT

Es gibt noch tausende weitere bekannte NP-vollständige Probleme.